PRIME HUNTING

Gábor Farkas[a*], Zsombor Kiss[b], Dániel Papatyi[b], Krisztina Schäffer[b]

[a] *ELTE, Faculty of Informatics, Department of Computer Algebra, associate professor*
[b] *ELTE, Computer Scientist BSc, II. year*

ABSTRACT

"Prime hunting" can be considered as a research area of computational number theory. Its goal is to find special combinations of integers and prove their primality. Four research groups, established by A. Járai between 1992 and 2014, published numerous world class scientific results. In this period, due to Járai's arithmetic routines fastest in the world, they reached the world record 19 times, namely found the largest known twin primes 9 times, Sophie Germain primes 7 times, a prime of the form $n^4 + 1$, a number which is simultaneously twin and Sophie Germain prime and the three largest known primes forming a Cunningham chain of length 3 of the first kind. When A. Járai retired, the investigations of this area were suspended. In the beginning of 2020 the research was reopened by G. Farkas at the newly-founded campus (Szombathely) of ELTE. The first signal success came in the end of May 2020. They proved the primality of the numbers which form the largest known Cunningham chains of length 2 of the 2nd kind. In this paper we report on a newly started prime hunting project with the aim of increasing our students' research activity.

Keywords: *computational number theory, prime tests, curious prime combinations, prime records*

1. Introduction

A well-organized collection of large prime numbers was started in 1984 when the name "Titanic Prime" was given to the primes with more than 1000 decimal digits by S. Yates [1]. We can say that a contest started for primes categorized by their "archivable" form. The exact definition of archivable form can be found on the site [2] which is supervised by C. K. Caldwell since 1996. For example, twin, Sophie Germain and Mersenne are well-known archivable forms. On Caldwell's site a top list can be seen for each form which contains the largest known primes in that category.

Reading about a world record prime, a question always arises as to why people want to find these numbers. In our point of view there are many reasons, for example knowing the newest scientific results and methods, designing and developing fast and effective programs, testing the hardware and continuing a beautiful tradition. From $\approx$ 300 BC to 2020 numerous famous mathematicians became "large prime collectors", for example Euclid, Descartes, Fermat, Mersenne, Leibniz, Euler, Lucas, Catalan, Cunningham, Pepin, Putnam, Lehmer and Járai.

2. Prime tests

A prime hunting process can be divided into two main parts, namely the sieving and primality checking. In this chapter we focus our attention to the theoretical background of prime tests.

*Corresponding author: Gábor Farkas, farkasg@inf.elte.hu
https://doi.org/10.37775/EIS.2020.2.1

2.1. Conclusive Prime Tests

Deciding whether a given number n is prime or not, is a problem that has interested mathematicians for ages. The oldest algorithm is more than 2500 years old, aptly called trial division, it simply checks if any number smaller than n divides it or not. The algorithm can be improved by noticing that one only needs to check for prime divisors less or equal than $\sqrt{n}$, however this is still extremely slow. Trial division, combined with other more efficient methods, is still used to this day for testing whether a number has small prime factors [3].

There are much faster modern algorithms – such as the AKS primality test – that is proven to run in polynomial time and ECPP, which can be proven to also run in polynomial time contingent on some conjectures [3]. Significantly better running times can be achieved with simpler algorithms if we only consider numbers of special forms. Two such tests are the *Reisel test* and the *Proth test.*

Proth test

The Proth test is a primality test for numbers of the form $N = k \cdot 2^n + 1$, where k is odd and $2^n > k$. These numbers are known as Proth numbers. The algorithm is based on the following theorem published in 1878 [4]:

Theorem 2.1 *(Proth's theorem) Let N be as described above, if there exists an integer a such that*

$$a^{(N-1)/2} \equiv -1 \pmod{N},$$

then N is prime.

There is no efficient strategy to find such an a, therefore the test's running time varies randomly.

Riesel test

Let $N = h \cdot 2^n - 1$, such that $2 \leq n \in \mathbb{N}$, $2^n > h \in \mathbb{Z}^+$ is odd, $D > 1$ a square-free positive integer, $a, b \in \mathbb{Z}$, $r = |a^2 - b^2 D|$, $\alpha = (a + b\sqrt{D})^2/r \in I_D$ the ring of algebraic integers (quadratic integers) of the quadratic field of $\mathbb{Q}(\sqrt{D})$, furthermore suppose that:

$$\left(\frac{D}{N}\right) = -1, \tag{1}$$

$$\left(\frac{r}{N}\right)(a^2 - b^2 D)/r = -1, \tag{2}$$

where $\left(\frac{D}{N}\right)$ is the Jacobi symbol, which is an extension of the Legendre symbol. Let m be an integer, p a prime, then the Legendre symbol is defined as:

$$\left(\frac{m}{p}\right) = \begin{cases} 0 & \text{if } p \mid m, \\ 1 & \text{if } p \nmid m \text{ and } x^2 \equiv m \pmod{p} \text{ has a solution,} \\ -1 & \text{if } p \nmid m \text{ and } x^2 \equiv m \pmod{p} \text{ has no solution.} \end{cases}$$

The Jacobi symbol is the unique extension of the Legendre symbol to all positive odd integers $n = b \cdot c$ with the multiplicative property:

$$\left(\frac{m}{n}\right) = \left(\frac{m}{b}\right) \cdot \left(\frac{m}{c}\right).$$

Now, we can formulate Riesel's famous assertion:

Theorem 2.2 *Suppose the above conditions hold, then N is prime if and only if*

$$u_{n-2} = \alpha^{h2^{n-2}} + \alpha^{-h2^{n-2}} \equiv 0 \pmod{N},$$

where $u_0 = \alpha^h + \alpha^{-h}$ and $u_i = u_{i-1}^2 - 2$.

H. Riesel published the proof in [5]. We show an example where we set $D = 13$, $a = 3$, and $b = 1$ as hyperparameters. Using a method described in [6] we can always find values h for which Eq. (1) holds. Eq. (2) always holds because $r = 4$, so

$$\left(\frac{r}{N}\right) = \left(\frac{2^2}{N}\right) = \left(\frac{2}{N}\right)^2$$

which can only equal 1 since N is odd.

We represented quadratic integers in the form

$$x + y \cdot \frac{(1 + \sqrt{13})}{2}$$

with x and y being integers. First we computed

$$\alpha^h = x + y \cdot \frac{(1 + \sqrt{13})}{2}$$

with a simple repeated-squaring algorithm implemented by us for quadratic integers, analogous to the one for rational integers in [7]. After that we stored

$$\alpha^h + \alpha^{-h} = 2 \cdot x + y$$

and used the identity $c^2 + c^{-2} = (c + c^{-1})^2 - 2$ for $c = \alpha^{h2^i}$ to compute

$$\alpha^{h2^{n-2}} + \alpha^{-h2^{n-2}}$$

working with rational integers instead of quadratic integers.

Since we only use this test to verify the primality of just a few numbers, extreme efficiency is not necessary. For example, in the case $n = 36\ 627$ and $h = 778\ 965\ 587\ 811$ our implementation was able to verify the number's primality in 7.72784 seconds.

2.2. Probabilistic primality tests

Probabilistic tests, as opposed to deterministic ones, do not always correctly answer whether a given number is prime or not, but they are often the preferred choice in practice, as they are significantly faster and there is only a small chance that they are mistaken (around one in a billion).

One of the simplest probabilistic tests is the *Fermat test,* in which we simply compute $a^{n-1} \pmod{n}$, for an arbitrary base a relatively prime to n. The result is 1 if n is prime, according to Fermat's Little Theorem, and is different from 1 with good chance if n is composite.

The *Fermat test* can be improved if we consider only odd candidates $n > 8$ and write them as

$$n = 1 + q \cdot 2^k,$$

where q is odd. If n is prime, once again applying Fermat's Little Theorem we get

$$a^{q \cdot 2^k} \equiv 1 \pmod{n}.$$

Upon further consideration, we can notice that if n is prime,

$$x^2 \equiv 1 \pmod{n}$$

if and only if

$$x^2 - 1 = (x+1)(x-1) \equiv 0 \pmod{n}.$$

Thus, $x \equiv 1$ or $x \equiv -1$, e.g. $a^{q \cdot 2^k} \equiv 1$ if and only if $a^q \equiv 1$ or $a^{q \cdot 2^j} \equiv -1$ for some $0 \leq j \leq k-1$, which can easily be checked during the computation of $a^{q \cdot 2^{k-1}} \pmod{n}$. This method is known as the *Miller–Rabin test* [3].

We compared the execution times of these algorithms for $a = 2$, and numbers of the form $h \cdot 2^m - 1$, because only these were considered in our prime hunting project. As expected, there was no noticeable difference, in fact since $h \cdot 2^m - 1 = 2 \cdot (h \cdot 2^{m-1} - 1) + 1$ only one extra comparison had to be made in the *Miller–Rabin test.*

The *Miller–Rabin test* can be converted into a deterministic test by testing for several bases, and applying a bound on the maximum number of bases that give false positives (these bases are known as liars). One such bound is $n/4$, which can be obtained by elementary considerations, but this results in a rather slow algorithm. There is a significantly better bound, but unfortunately it relies on the Generalised Riemann Hypothesis being true, so the *Miller–Rabin test* is not used as a deterministic test.

Theorem 2.3 *Let $n > 8$ be a non-prime power composite number, if the Generalised Riemann Hypothesis is true, then there exists a base*

$$a < 2(\ln(n))^2$$

with which the Miller–Rabin-test discovers that n is composite.

Algorithm 1 The pseudocode of the Miller–Rabin primality test

```
1: procedure MillerRabin(n)                          ▷ Miller–Rabin primality test
2:     a ← 2
3:     k ← log_2(n − 1)/2
4:     j ← k
5:     b ← a^q (mod n)
6:     if (((j = k) and (b = 1)) or (b = n − 1)) then
7:         return probably prime
8:     end if
9:     if ((j < k) and (b = 1)) then
10:        return composite
11:    end if
12:    j ← j − 1
13:    if (j > 0) then
14:        b ← b^2 (mod n)
15:        goto 6
16:    end if
17:    return composite
18: end procedure
```

For the above mentioned prime tests Z. Kiss and D. Papatyi gave their own implementation in our project. We plan to use these routines in a prime hunting project in the near future, therefore we are working on further improving the efficiency of these programs. The pseudocode of the *Miller–Rabin test* can be seen in Algorithm 1.

3. Prime hunting

Since our prospective purpose is to reach such results as Járai et al. published in [8-21], apart from primality tests, we also have to implement some efficient sieving programs. In the following part of this paper the work described in [9] is called $Cc22-project$.

We focus on the following archivable forms: twin primes, Sophie Germain primes and Cunningham chains.

Definition 3.1 *Let p be a prime. If*

- *$p+2$ is also a prime then p and $p+2$ are **twin primes**,*
- *$2p+1$ is also a prime then p is a **Sophie Germain prime**,*
- *$2p+1$ and $4p+3$ are primes then p is a **double Sophie Germain or safe prime**,*
- *if we have a sequence of primes*

$$\{p, 2p+d, 4p+3d, 8p+7d, ..., 2^{k-1}p+(2^{k-1}-1)d\},$$

*where $k \geq 2$, we speak about a **Cunningham chain of length k of the first kind** (notation: $Cck1$) if $d=1$ and a **Cunningham chain of length k of the second kind** (notation: $Cck2$) if $d=-1$.*

Before programming we need to calculate the value of some important hyperparameters. First of all, we choose a set $H = \{1, 2, \ldots, 2^R - 1\}$. To find the value of R the Bateman–Horn conjecture [22] is used. The exact computations of the hyperparameters were published in [23] and [24]. Afterwards we generate sequences of large numbers by substituting the elements of H into *generator polynomials*. These sequences include the prime numbers searched for.

3.1. Generator polynomials

A generator polynomial can be given in the form:

$$f_i(x) = (h_0 + c \cdot x) \cdot 2^{e+i} + j, \tag{3}$$

where j is equal 1 or -1 and $i \in \{0, 1, \ldots, l\}$. The description of the integers e, h_0 and c can be seen later in this subsection. The numbers generated by a generator polynomial are called *candidates*. Consider the following deductions: if we set $j=1$ then for the polynomials

$$f_i(x) = (h_0 + c \cdot x) \cdot 2^{e+i} + 1,$$

where $i = 0, 1$, the equation $f_1 = 2f_0 - 1$ is valid because

$$(h_0 + c \cdot x) \cdot 2^{e+1} + 1 = 2 \cdot ((h_0 + c \cdot x) \cdot 2^e + 1) - 1.$$

Thus, if for an arbitrary $x \in H$, $f_0(x)$ and $f_1(x)$ are simultaneously prime then they form a $Cc22$.

We can observe easily that the polynomials Eq. (3) can generate a $Cck2$ for an arbitrary k if $i = 0, 1, \ldots, k$ and $f_0(x), f_1(x), \ldots, f_k(x)$ are simultaneously primes for some $x \in H$. Furthermore, we can generate sequences for $Cck1$ too if we use the value $j=-1$.

Algorithm 2 The pseudocode of the sieving method

procedure SIEVE(R, p_S, e, h_0, c) ▷ Sieve
  $B[0] \leftarrow 1, B[1] \leftarrow 1, \ldots, B[|H|-1] \leftarrow 1$
  Produce "small primes" P_S: primes up to p_S with sieve of Eratosthenes
  $p \leftarrow$ the smallest prime in P_S
  while $p \leq p_S$ **do**
    for $i = 1$ to k **do**
      Let h be the solution of the congruence $f_i(x) \equiv 0 \pmod{p}$
      $B[h] \leftarrow 0, B[h+p] \leftarrow 0, \ldots, B[h+q \cdot p] \leftarrow 0,$ where $(h + q \cdot p) < |H|, q \in \mathbb{N}$
    end for
    $p \leftarrow nextprime(p)$
  end while
  $A \leftarrow$ {the indexes of elements 1 in B}
  Do in parallel
    Compute thread 1
    Compute thread 2
      ⋮
    Compute thread n
  EndDo
  $\overline{A} \leftarrow \bigcap_{j=1}^{n} A_j$
  return $\overline{A}$
end procedure

If we hunt for twin primes the appropriate generator polynomials are

$$f_i(x) = (h_0 + c \cdot x) \cdot 2^e + (-1)^{i+1},$$

where $i = 0, 1$.

Consider now the other hyperparameters. The constant h_0 guarantees that the generated numbers satisfy the Lucasian criteria for the Riesel primality test shown in the previous section. The constant e is called the exponent and it sets the magnitude of the generated numbers. The constant c is the product of the first k prime numbers:

$$c = p_1 \cdot p_2 \cdot \ldots p_k,$$

where $k \in \mathbb{N}^+$ and p_i is the i-th prime. If we correctly set the values of the parameters e, h_0 and c then the numbers generated by polynomials of the form Eq. (3) are sufficiently large, satisfy the conditions of the Riesel test and do not have prime factors $p|c$.

If the goal is to generate numbers of the form $k \cdot 2^n + 1$ then $h_0 = 0$ can be chosen. For example, Farkas et al in $Cc22 - project$ set the parameters in the following way: $h_0 = 0$, $c = 30030$ and $e = 256000$. Thus, the generated numbers had no prime factors less than 17 and had at least 77 069 decimal digits.

3.2. Sieving

The primality tests are highly time-consuming therefore the number of candidates must be reduced. In order to explain our method, we introduce some new concepts. Consider the set $P_S = \{p_{k+1}, p_{k+2}, \ldots, p_S\}$ which contains all prime numbers between p_{k+1}, the smallest prime which

Algorithm 3 The pseudocode of thread i

procedure THREAD $i(A, p_a, p_b)$ ▷ p_a, p_b are primes and $p_S < p_a < p_b < p_S^2$
  $A_i \leftarrow A$
  Produce "small primes" P_S: primes up to p_S with sieve of Eratosthenes
  while $\exists p \in P_L$ where p has not yet been used as a sieving prime **do**
    while $p < p_b$ **do**
      Produce the primes from p_a to $p_b \rightarrow PST$ with sieve of Eratosthenes
      $p \leftarrow$ the smallest element of PST
      for $j = 1$ to k **do**
        Let h be the solution of the congruence $f_j(x) \equiv 0 \pmod{p}$
        $A_i \leftarrow A_i \backslash \{h, h + 2p, \ldots, h + q \cdot p < |H|\}$
      end for
      $p \leftarrow nextprime(p)$
    end while
    Update p_a and p_b
  end while
  return A_i
end procedure

is larger than the maximal factor of c, and a given prime p_S. The elements of P_S are called *small sieving primes*. We say that a prime p is a *large sieving prime* if $p_S < p \leq p_S^2$ holds.

In our project by **sieving** we mean a method which eliminates the elements from H which generate numbers having a prime factor up to $p_M \approx p_S^2$. The exact description of this method can be found in subsection 3.3.

We have to point out that the more than 2000 years old method called *sieve of Eratosthenes* is still a cutting edge tool for finding all prime numbers up to a given bound even in 2020. Of course, a time-efficient implementation is necessary. For example, the program run in the $Cc22 - project$ produced and stored in an array the primes up to $p_S = 134\ 217\ 689 \approx 2^{27}$ in less than 5 seconds.

3.3. The details of the sieving method

As a matter of fact the our sieving is a simple variant of the generalized sieve with some linear polynomials $f_i(x)$ where $i = 1, 2, \ldots, k$.

If we set the parameters correctly, then for every $p > p_k$ the linear congruence $f_i(x) \equiv 0 \mod p$ must have exactly one incongruent solution, which is denoted by h. It is clear that the numbers $h, h + p, h + 2p, ... \in H$ generate numbers having a prime factor p. Now we can say that "we sieve H with p", i.e. we remove the set $\{h, h + p, h + 2p, ...\}$ from H. Naturally, each sieving prime can sieve at most k-times. In practice, we defined an array B such that the elements of H are the indexes of B. Every value in B is initially 1. If we find that any $f_i(h)$ has a prime factor in P_S for an $h < |H|$, we change the values $B[h], B[h + p], B[h + 2p], \ldots, B[h + qp]$ to 0, for all $h + pq < |H|$. The sieve with small primes can be carried out easily on one fast processor with large operative memory, but the large prime sieve needs numerous processors and threads. In order to be able to continue the sieving on lower performance computers, we store the indexes of the 0 elements into an array A. A is about 30 times smaller than B, so we can take advantage of the cache memory and we can make the parallelization of the further sieving even with lower-capacity processors.

Now assume that we have a multiprocessor computer or a grid and we can simultaneously use n threads and let us consider the set of large primes, i. e. $P_L = \{p \in \mathcal{P} \wedge p_S < p < p_S^2\}$, where $\mathcal{P}$

denotes the set of prime numbers. Every thread receives a copy of A, a lower and an upper bound from the interval $(p_S, p_S^2]$. Denote by A_i the copy of A received by the ith thread ($1 \leq i \leq n$). Then every thread performs a sieving process from the given lower bound to the upper bound, producing an array of primes, a so called prime sieve table. Then the thread sieves its own copy of A with the primes found in its own prime sieve table. In this stage the effect of sieving on the set A can be described by the following command:

$$A := A \setminus \{h, h+p, h+2p, ...h+qp | q \in \mathbb{N}, h+qp \leq |H|\}.$$

After the ith thread finished sieving with all primes in its prime sieve table, A will be updated which means that we remove all numbers x from A for which $x \notin A_i$. Then if necessary the prime sieve table will be refilled by large primes and the sieving of the updated A_i may be restarted.

We stop sieving if our program completes the sieving for all primes $p \leq p_S^2$. Finally, the output set (or vector) $\overline{A}$ contains numbers which have not prime factors less than p_S^2. The pseudocode of our complete sieving method can be seen in Algorithm 2 and Algorithm 3.

4. Conclusions

As the test results our programs show, in the nearly future an effective implementation of the probability primality tests combined our newly designed "generalized sieve algorithm" would be an appropriate tool for investigating large "curious prime combinations".

5. Acknowledgement

The project has been supported by the European Union, cofinanced by the European Social Fund, EFOP-3.6.1.-16-2016-0023.

6. References

[1] S. Yates, *Titanic primes*, Journal of Recreational Mathematics 16(4), 1984, pp. 250-262.

[2] C.K. Caldwell, *The Prime Pages*, 2020, url

[3] S.Y. Yan, *Computational Number Theory and Modern Cryptography*, Higher Education Press, 2013, ISBN 978-1-118-18858-3

[4] P. Ribenboim, *The Little Book of Bigger Primes*, Springer, 2004, ISBN 0-387-20169-6

[5] H. Riesel, *Lucasian Criteria for the Primality of* $N = h \cdot 2^n - 1$, Math. Comp. 23, 1969, pp. 869-875. CrossRef

[6] G. Farkas, *Quadratic Fields from Mathematics and Informatics Point of View*, *Habilitation Thesis*, ELTE-IK Budapest, 2012, 109 p.

[7] V. Shoup, *A Computational Introduction to Number Theory and Algebra*, Cambridge University Press, 2008, CrossRef

[8] K.H. Indlekofer - A. Járai, *Largest Known Twin Primes and Sophie Germain Primes*, Mathematics of Computation 68(227), 1999, pp. 1317-1324. CrossRef

[9] G. Farkas, G. Gévay, P. Magyar, B. Szekeres, *Járai's Prime Hunting Method Reloaded (The Largest Known Cunningham Chain of Length 2 of the 2nd Kind)*, Annales Univ. Sci. Budapest, Sect. Comp. 50, 2020, accepted.

[10] K.H. Indlekofer - A. Járai, *Largest Known Twin Primes*, Mathematics of Computation 65, 1996, pp. 427-428. CrossRef

[11] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *Report on the largest known twin primes*, Annales Univ. Sci. Budapest, Sect. Comp. 25, 2005, pp. 247–248.

[12] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *The largest known twin primes of the World*, $16869987339975 \cdot 2^{171960} \pm 1$ (51779 *digits*), 2005, url

[13] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *Report on the largest known Sophie Germain and twin primes*, Annales Univ. Sci. Budapest, Sect. Comp. 26, 2006, pp. 181-183.

[14] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *The largest known twin primes of the World*, $100314512544015 \cdot 2^{171960} \pm 1$ (51780 *digits*), 2006, url

[15] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *The largest known Sophie Germain prime of the World*, $137211941292195 \cdot 2^{171960} - 1$ (51780 *digits*), 2006, url

[16] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *The largest known twin primes of the World*, $194772106074315 \cdot 2^{171960} \pm 1$ (51780 *digits*), 2007, url

[17] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *The largest known Sophie Germain prime of the World*, $620366307356565 \cdot 2^{253824} - 1$ (76424 *digits*), 2009, url

[18] T. Csajbók, G. Farkas, A. Járai, Z. Járai, J. Kasza, *The largest known Sophie Germain prime of the World*, $648621027630345 \cdot 2^{253824} - 1$ (76424 *digits*), 2009, url

[19] G. Farkas, G. Gévay, A. Járai, E. Vatai, *The largest known Cunningham chain of length 3 of the first kind*, Studia Universitatis Babes-Bolyai Mathematica 59(4), 2014, pp. 457-462.

[20] A. Járai, *World Records in Computational Number Theory*, url

[21] A. Járai, *Számítógépes számelmélet*, 2009, url

[22] P.T. Bateman, R.A. Horn, *A heuristic asymptotic formula concerning the distribution of prime numbers*, Mathematics of Computation 16(79), 1962, pp. 363-367. CrossRef

[23] T. Csajbók, G. Farkas, J. Kasza, *Sieving for Large Twin Primes and Cunningham chains of length 2 of the second kind*, Annales Univ. Sci. Budapest, Sect. Comp. 38, 2012, pp. 117-128.

[24] G. Farkas, E. Vatai, *Sieving for Large Cunningham Chains of Length 3 of the First Kind*, Annales Univ. Sci. Budapest, Sect. Comp. 40, 2013, pp. 215-222.